Lossy Encryption from General Assumptions

Brett Hemenway and Rafail Ostrovsky

Crypto in the Clouds Workshop, MIT

August 5, 2009
Outline

Motivation

Definitions

Our Results
Outline

Motivation

Definitions

Our Results
Motivation
Motivation

Do the uncorrupted messages remain secure?
Motivation

Do the uncorrupted messages remain secure?

\[ e_i = E(pk, m_i, r_i) \]
Motivation

\[ e_i = E(pk, m_i, r_i) \]
Motivation

\[ e_i = E(pk, m_i, r_i) \]
Motivation

Do the uncorrupted messages remain secure?

Brett Hemenway and Rafail Ostrovsky
Motivation

Do the uncorrupted messages remain secure?

\[ e_i = E(pk, m_i, r_i) \]
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97, CHK05)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/ Meaningless Encryption (KN08)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97, CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)
- Lossy Encryption (BHY09)
Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97, CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/Meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)
- Lossy Encryption (BHY09)
Outline

Motivation

Definitions

Our Results
Selective Opening Security

This type of security is called Selective Opening Security. Recognized long ago in folklore. Formalized in [DNRS03],[BHY09]. If the adversary does not learn the randomness, then this follows from IND-CPA security. If the messages are independent, then this follows from IND-CPA security. No one has been able to show that IND-CPA security implies IND-SOA security. No one has been able to exhibit an IND-CPA secure system that is not IND-SOA security.
Selective Opening Security

- This type of security is called Selective Opening Security.
Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
This type of security is called Selective Opening Security.

- Recognized long ago in folklore.
- Formalized in [DNRS03],[BHY09]
Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
- If the messages are independent, then this follows from IND-CPA security.
This type of security is called Selective Opening Security.

- Recognized long ago in folklore.
- Formalized in [DNRS03],[BHY09]

If the adversary does not learn the randomness, then this follows from IND-CPA security.

If the messages are independent, then this follows from IND-CPA security.

No one has been able to show that IND-CPA security implies IND-SOA security.
Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
- If the messages are independent, then this follows from IND-CPA security.
- No one has been able to show that IND-CPA security implies IND-SOA security.
- No one has been able to exhibit an IND-CPA secure system that is not IND-SOA security.
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

\[ \text{IND-SO-ENC (Ideal)} \]
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

▶ \((m_1, \ldots, m_n) \leftarrow M\)
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

1. $(m_1, \ldots, m_n) \leftarrow M$
2. $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
3. $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n)))$
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- \((m_1, \ldots, m_n) \leftarrow M\)
- \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
- \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n)))\)
- \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))\)
Selective Opening Security: Indistinguishability [BHY09]

**IND-SO-ENC (Real)**

1. \((m_1, \ldots, m_n) \leftarrow M\)
2. \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
3. \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
4. \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))\)

**IND-SO-ENC (Ideal)**
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

1. $(m_1, \ldots, m_n) \leftarrow M$
2. $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
3. $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
4. $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

1. $(m_1, \ldots, m_n) \leftarrow M$
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

1. \((m_1, \ldots, m_n) \leftarrow M\)
2. \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
3. \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
4. \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))\)

IND-SO-ENC (Ideal)

1. \((m_1, \ldots, m_n) \leftarrow M\)
2. \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
3. \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
4. \((m'_1, \ldots, m'_n) \leftarrow M|M_I\)
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- \((m_1, \ldots, m_n) \leftarrow M\)
- \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
- \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
- \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))\)

IND-SO-ENC (Ideal)

- \((m_1, \ldots, m_n) \leftarrow M\)
- \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
- \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
- \((m'_1, \ldots, m'_n) \leftarrow M|M_I\)
- \((m'_1, \ldots, m'_n) \leftarrow M|M_I\)
- \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m'_1, \ldots, m'_n))\)
Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

1. \((m_1, \ldots, m_n) \leftarrow M\)
2. \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
3. \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
4. \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))\)

IND-SO-ENC (Ideal)

1. \((m_1, \ldots, m_n) \leftarrow M\)
2. \(r_1, \ldots, r_n \leftarrow \text{coins}(E)\)
3. \(I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))\)
4. \((m_1', \ldots, m_n') \leftarrow M | M_I\)
5. \((m_1, \ldots, m_n) \leftarrow M | M_I\)
6. \(b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1', \ldots, m_n'))\)

\[\left| \Pr[A^{IND-SO-ENC-REAL} = 1] - \Pr[A^{IND-SO-ENC-IDEAL} = 1] \right| < \nu\]
Lossy Encryption in Detail

\[ G(1^\lambda, \text{mode}), E(pk, m, r), D(sk, c) \]

**Correctness:**
For all \( m, r \)
\[ D(E(pk_I, m, r)) = m \]

**Lossiness:**
For all \( m_0, m_1 \)
\[ \{ E(pk_L, m_0, r) \} \approx_s \{ E(pk_L, m_1, r) \} \]

**Indistinguishability**
\[ \{ pk_I : pk_I \leftarrow G(1^\lambda, \text{Injective}) \} \approx_c \{ pk_L : pk_L \leftarrow G(1^\lambda, \text{Lossy}) \} \]
Lossy Encryption in Detail

$G(1^\lambda, \text{mode}), E(pk, m, r), D(sk, c)$

**Correctness:**

For all $m, r$

$D(E(pk_I, m, r)) = m$

**Lossiness:**

For all $m_0, m_1$

$\{ E(pk_L, m_0, r) \} \approx^s \{ E(pk_L, m_1, r) \}$

**Indistinguishability**

$\{ pk_I : pk_I \leftarrow G(1^\lambda, \text{Injective}) \} \approx^c \{ pk_L : pk_L \leftarrow G(1^\lambda, \text{Lossy}) \}$
Lossy Encryption in Detail

\[ G(1^\lambda, \text{mode}), E(pk, m, r), D(sk, c) \]

**Correctness:**
For all \( m, r \)
\[ D(E(pk_I, m, r)) = m \]

**Lossiness:**
For all \( m_0, m_1 \)
\[ \{ E(pk_L, m_0, r) \} \approx_s \{ E(pk_L, m_1, r) \} \]

**Indistinguishability**
\[ \{ pk_I : pk_I \leftarrow G(1^\lambda, \text{Injective}) \} \approx_c \{ pk_L : pk_L \leftarrow G(1^\lambda, \text{Lossy}) \} \]
Lossy Encryption in Detail

\[ G(1^\lambda, \text{mode}), E(pk, m, r), D(sk, c) \]

**Correctness:**
For all \( m, r \)
\[ D(E(pk_I, m, r)) = m \]

**Lossiness:**
For all \( m_0, m_1 \)
\[ \{ E(pk_L, m_0, r) \} \approx^s \{ E(pk_L, m_1, r) \} \]

**Indistinguishability**
\[ \{ pk_I : pk_I \leftarrow G(1^\lambda, \text{Injective}) \} \approx^c \{ pk_L : pk_L \leftarrow G(1^\lambda, \text{Lossy}) \} \]
Lossy Encryption in Detail

\[ G(1^\lambda, \text{mode}), E(pk, m, r), D(sk, c) \]

**Correctness:**
For all \( m, r \)
\[ D(E(pk_I, m, r)) = m \]

**Lossiness:**
For all \( m_0, m_1 \)
\[ \{ E(pk_L, m_0, r) \} \approx^s \{ E(pk_L, m_1, r) \} \]

**Indistinguishability**
\[ \{ pk_I : pk_I \leftarrow G(1^\lambda, \text{Injective}) \} \approx^c \{ pk_L : pk_L \leftarrow G(1^\lambda, \text{Lossy}) \} \]

Notice: Indistinguishability + Lossiness \( \implies \) IND-CPA security
Lossy Encryption is IND-SO-ENC Secure (BHY09)

In Lossy mode, the distributions

\[(E(m_1, r_1), \ldots, E(m_n, r_n)) \approx^s (E(m'_1, r_1), \ldots, E(m'_n, r_n))\]

Since the encryptions are statistically independent of the messages, so even after conditioning on certain openings, the rest remain independent of the messages.
ReRandomizable Encryption

\[ (G, E, D) \text{ is semantically secure}. \]

There exists a function \( \text{ReRand} \) such that for all \( \text{pk}, m, r \),

\[ \text{Correctness: } D(\text{ReRand}(E(\text{pk}, m, r))) = m \]

\[ \text{Statistical rerandomization: } \{ \text{ReRand}(E(\text{pk}, m, r)) \} \approx \{ \text{ReRand}(E(\text{pk}, m, r')) \} \]

Brett Hemenway and Rafail Ostrovsky
ReRandomizable Encryption

- $(G, E, D)$ is semantically secure.
ReRandomizable Encryption

- $(G, E, D)$ is semantically secure.
- There exists a function $\text{ReRand}$ such that for all $pk, m, r, r'$
ReRandomizable Encryption

- \((G, E, D)\) is semantically secure.
- There exists a function \(\text{ReRand}\) such that for all \(pk, m, r, r'\)
  - Correctness:

\[
D(\text{ReRand}(E(pk, m, r))) = m
\]
ReRandomizable Encryption

- $(G, E, D)$ is semantically secure.
- There exists a function ReRand such that for all $pk, m, r, r'$
  - Correctness:
    \[
    D(\text{ReRand}(E(pk, m, r))) = m
    \]
  - Statistical rerandomization:
    \[
    \{\text{ReRand}(E(pk, m, r))\} \approx^s \{\text{ReRand}(E(pk, m, r'))\}
    \]
Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$
Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Caution: this is not necessarily statistically re-randomizing.
Homomorphic Encryption

If \( E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*), \) then we can re-randomize by doing

\[
\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').
\]

Caution: this is not necessarily statistically re-randomizing.

It is statistically re-randomizing for all known homomorphic cryptosystems.
Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Caution: this is not necessarily statistically re-randomizing.

It is statistically re-randomizing for all known homomorphic cryptosystems.

If you can sample statistically close to uniformly from the set of encryptions of 0 then homomorphic encryption is statistically rerandomizable.
Outline

Motivation

Definitions

Our Results
Our Results

ReRandomizable Encryption "is" Lossy Encryption

A framework for creating Lossy Encryption:

Applying the results of [BHY09] gives:

Goldwasser-Micali
El-Gamal
Paillier / Damg˚ ard-Jurik

The first proof that Paillier/Damg˚ ard-Jurik is SEM-SO-ENC secure.

Statistically Hiding-OT implies Lossy Encryption
PIR implies Lossy Encryption
Homomorphic Encryption implies Lossy Encryption

CCA2 Selective Opening Secure definitions and constructions
Constructions from statistically-hiding NIZKs in the simulation-based model
Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damg˚ ard-Jurik

The first proof that Paillier/Damg˚ ard-Jurik is SEM-SO-ENC secure. This is the most efficient known SEM-SO-ENC cryptosystem.

- Statistically Hiding-OT implies Lossy Encryption
- PIR implies Lossy Encryption
- Homomorphic Encryption implies Lossy Encryption

- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik

- The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
  - This is the most efficient known SEM-SO-ENC cryptosystem.

- Statistically Hiding-OT implies Lossy Encryption
- PIR implies Lossy Encryption
- Homomorphic Encryption implies Lossy Encryption

- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
Our Results

▶ ReRandomizable Encryption “is” Lossy Encryption
  ▶ A framework for creating Lossy Encryption:
  ▶ Applying the results of [BHY09] gives:
    ▶ Goldwasser-Micali
    ▶ El-Gamal
    ▶ Paillier / Damgård-Jurik
  ▶ The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
▶ Statistically Hiding-OT implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.

- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption

- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
ReRandomizable Encryption “is” Lossy Encryption

\[ \text{Let } (G, E, D, \text{ReRand}) \text{ be a ReRandomizable Encryption.} \]

\[ \text{Let } (pk, sk) \leftarrow G_{e0} = E(pk, b_0, r_0), e_1 = E(pk, b_1, r_1). \]

Define \( PK = (pk, e_0, e_1), SK = sk. \)

\[ \text{Encryption of } b \text{ will be } \text{ReRand}(e_b). \]

\[ \text{Decryption is the same as for the ReRandomizable scheme.} \]

This is lossy if \( b_0 = b_1 \), and injective if \( b_0 \neq b_1 \).

The indistinguishability of modes follows immediately from the Semantic Security of \( (G, E, D) \).
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
- Let \((pk, sk) \leftarrow G\)
  
  \[ e_0 = E(pk, b_0, r_0), \quad e_1 = E(pk, b_1, r_1). \]

  Define \(PK = (pk, e_0, e_1), \ SK = sk.\)
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
- Let \((pk, sk) \leftarrow G\)
  \[ e_0 = E(pk, b_0, r_0), \quad e_1 = E(pk, b_1, r_1). \]
  Define \(PK = (pk, e_0, e_1), \ SK = sk.\)
- Encryption of \(b\) will be \(\text{ReRand}(e_b).\)
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
- Let \((pk, sk) \leftarrow G\)
  \[
e_0 = E(pk, b_0, r_0), \quad e_1 = E(pk, b_1, r_1).
\]
  Define \(PK = (pk, e_0, e_1), \ SK = sk\).
- Encryption of \(b\) will be
  \[
  \text{ReRand}(e_b).
  \]
- Decryption is the same as for the ReRandomizable scheme.
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
- Let \((pk, sk) \leftarrow G\)
  \[ e_0 = E(pk, b_0, r_0), \quad e_1 = E(pk, b_1, r_1). \]
  Define \(PK = (pk, e_0, e_1), \) \(SK = sk.\)
- Encryption of \(b\) will be \(\text{ReRand}(e_b).\)

- Decryption is the same as for the ReRandomizable scheme.

This is lossy if \(b_0 = b_1\), and injective if \(b_0 \neq b_1\).
ReRandomizable Encryption “is” Lossy Encryption

- Let \((G, E, D, \text{ReRand})\) be a ReRandomizable Encryption.
- Let \((pk, sk) \leftarrow G \quad e_0 = E(pk, b_0, r_0), \quad e_1 = E(pk, b_1, r_1).\)
  Define \(PK = (pk, e_0, e_1), \quad SK = sk.\)
- Encryption of \(b\) will be \(\text{ReRand}(e_b).\)
- Decryption is the same as for the ReRandomizable scheme.

This is lossy if \(b_0 = b_1,\) and injective if \(b_0 \neq b_1.\)

The indistinguishability of modes follows immediately from the Semantic Security of \((G, E, D).\)
For Homomorphic Encryption

If \((G, E, D)\) is homomorphic and \(E(pk, 0, r)\) is statistically close to uniform on the set of encryptions of 0, then we can make lossy encryption, simply by setting \(PK = (pk, e)\) where \(e = E(pk, 0, r)\) in Lossy Mode and \(E(pk, 1, r)\) in injective mode.

Encryption of \(m\) is just \(e m \cdot E(pk, 0, r)\).

Decryption is the same.
If \((G, E, D)\) is homomorphic and \(E(pk, 0, r)\) is statistically close to uniform on the set of encryptions of 0, then.encryption of \(m\) is just \(e^m \cdot E(pk, 0, r)\). Decryption is the same.
For Homomorphic Encryption

- If $(G, E, D)$ is homomorphic and $E(pk, 0, r)$ is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting $PK = (pk, e)$ where $e = E(pk, 0, r)$ in Lossy Mode and $E(pk, 1, r)$ in injective mode.
For Homomorphic Encryption

- If \((G, E, D)\) is homomorphic and \(E(pk, 0, r)\) is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting \(PK = (pk, e)\) where \(e = E(pk, 0, r)\) in Lossy Mode and \(E(pk, 1, r)\) in injective mode.
- Encryption of \(m\) is just \(e^m \cdot E(pk, 0, r)\).
For Homomorphic Encryption

- If \((G, E, D)\) is homomorphic and \(E(pk, 0, r)\) is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting \(PK = (pk, e)\) where \(e = E(pk, 0, r)\) in Lossy Mode and \(E(pk, 1, r)\) in injective mode.
- Encryption of \(m\) is just \(e^m \cdot E(pk, 0, r)\).
- Decryption is the same.
Oblivious Transfer Implies Lossy Encryption

- Sender
- Receiver

Computational receiver privacy implies indistinguishability of modes
Statistical sender privacy implies lossiness of lossy branch
Oblivious Transfer Implies Lossy Encryption

\[ x_0 \quad x_1 \]

\[
\begin{array}{c}
\text{Sender} \\
\end{array}
\]

\[
\begin{array}{c}
\text{Receiver} \\
\end{array}
\]

\[ b \]

\[ E(m, r) \equiv Q^b(m, 0; r) \]

Computational receiver privacy implies indistinguishability of modes

Statistical sender privacy implies lossiness of lossy branch

Brett Hemenway and Rafail Ostrovsky
Oblivious Transfer Implies Lossy Encryption

\[ \text{Sender} \xrightarrow{\mathbf{x}_0, \mathbf{x}_1} Q_b(\cdot, \cdot; \cdot) \xrightarrow{} \text{Receiver} \]

PK_{\text{inj}}: Q_0 \xrightarrow{} PK_{\text{lossy}}: Q_1 \xrightarrow{\mathbf{m}, r} E(\mathbf{m}, r) \equiv Q_b(\mathbf{m}, 0; r)

Computational receiver privacy implies indistinguishability of modes
Statistical sender privacy implies lossiness of lossy branch

Brett Hemenway and Rafail Ostrovsky
Oblivious Transfer Implies Lossy Encryption

$Q_b(x_0, x_1; r) \\ Q_b(\cdot, \cdot; \cdot) \\ b$

Computational receiver privacy implies indistinguishability of modes
Statistical sender privacy implies lossiness of lossy branch

Brett Hemenway and Rafail Ostrovsky
Oblivious Transfer Implies Lossy Encryption

\[ Q_b(x_0, x_1; r) \]

Computational receiver privacy implies indistinguishability of modes
Statistical sender privacy implies lossiness of lossy branch

\[ PK_{inj}: \]
\[ Q_0 \]

\[ PK_{lossy}: \]
\[ Q_1 \]

\[ E(m, r) \equiv Q_b(m, 0; r) \]
Oblivious Transfer Implies Lossy Encryption

\[ x_0 \quad x_1 \quad Q_b(\cdot, \cdot; \cdot) \quad Q_b(x_0, x_1; r) \quad b \]

\[ PK_{inj}: \quad Q_0 \quad \quad \quad \quad \quad \quad PK_{lossy}: \quad Q_1 \]

\[ E(m, r) \equiv Q_b(m, 0; r) \]

Computational receiver privacy implies indistinguishability of modes
Statistical sender privacy implies lossiness of lossy branch

Brett Hemenway and Rafail Ostrovsky
Chosen Ciphertext Security

Chosen Ciphertext Security in the Selective Opening Setting
IND-SO-CCA2: Definitions

Challenger

Adversary
IND-SO-CCA2: Definitions

Challenger  Adversary

Decryption Queries

$D(c)$

Decryption Queries

Output $b$
IND-SO-CCA2: Definitions

**Challenger**

**Adversary**

**Decryption Queries**

**Selective Opening Query**
IND-SO-CCA2: Definitions

- Challenger
- Adversary

Decryption Queries

Selective Opening Query

Decryption Queries

Output $b$
IND-SO-CCA2: Definitions

Challenger

\[ c \]

\[ D(c) \]

Adversary

\[ \vdots \]

Selective Opening Query

Decryption Queries

Brett Hemenway and Rafail Ostrovsky
IND-SO-CCA2: Definitions

Challenger

C
D(C)

: : :

E(m₁, r₁), ..., E(mᵢ, rᵢ)

I

{lᵢ, rᵢ}ᵢ∈I, {m'ⱼ}ⱼ∉I

Decryption Queries

Adversary
IND-SO-CCA2: Definitions

Challenger

\[ c \]
\[ D(c) \]
\[ \vdots \]
\[ E(m_1, r_1), \ldots, E(m_n, r_n) \]
\[ l \]
\[ \{m_i, r_i\}_{i \in I}, \quad \{m'_j\}_{j \notin I} \]
\[ \vdots \]
\[ c \]
\[ D(c) \]

Adversary

Output \( b \)
Lossy Trapdoor Functions [PW08]

Injective Mode

Lossy Mode

$F_I \approx F_\ell$
Lossy Trapdoor Functions in Detail

\[(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)\]
Lossy Trapdoor Functions in Detail

\((s, t) \leftarrow G_{LTDF}(1^\lambda, inj)\) \quad (s, \bot) \leftarrow G_{LTDF}(1^\lambda, lossy)
Lossy Trapdoor Functions in Detail

\((s, t) \leftarrow \text{GLTDF}(1^\lambda, inj)\) \hspace{2cm} \((s, \perp) \leftarrow \text{GLTDF}(1^\lambda, \text{lossy})\)

Trapdoor:
\[ F^{-1}(t, F(s, x)) = x \]
Lossy Trapdoor Functions in Detail

\[(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)\]  
\[(s, \perp) \leftarrow G_{LTDF}(1^\lambda, lossy)\]

**Trapdoor:**
\[F^{-1}(t, F(s, x)) = x\]

**Lossiness:**
\[|imF(s, \cdot)| \leq 2^r\]
Lossy Trapdoor Functions in Detail

\[(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)\]

\[(s, \bot) \leftarrow G_{LTDF}(1^\lambda, lossy)\]

\[\text{Trapdoor: } F^{-1}(t, F(s, x)) = x\]

\[\text{Lossiness: } |imF(s, \cdot)| \leq 2^r\]

The first outputs of \(G_{LTDF}(1^\lambda, inj)\), and \(G_{LTDF}(1^\lambda, lossy)\) are computationally indistinguishable.
All-But-One Functions [PW08]

\[(s, t) \leftarrow G_{ABO}(1^\lambda, b^*)\]

**Trapdoor:**
For \(b \neq b^*\)

\[F^{-1}(t, b, F(s, b, x)) = x\]

**Lossiness:**
\[|imF(s, b^*, \cdot)| \leq 2^r\]

The first outputs of \(G_{ABO}(1^\lambda, b_0)\), and \(G_{ABO}(1^\lambda, b_1)\) are computationally indistinguishable.
All-But-$n$ Functions

$$(s, t) \leftarrow G_{ABN}(1^{\lambda}, B) \quad \text{with} \quad |B| = n$$

**Trapdoor:**

For $b \not\in B$

$$F^{-1}(t, b, F(s, b, x)) = x$$

**Lossiness:**

For $b \in B$

$$|imF(s, b, \cdot)| \leq 2^r$$

The first outputs of $G_{ABN}(1^{\lambda}, B_0)$, and $G_{ABN}(1^{\lambda}, B_1)$ are computationally indistinguishable.
All-But-\(n\) Functions

\[(s, t) \leftarrow G_{ABN}(1^\lambda, B) \quad \text{with } |B| = n\]

**Trapdoor:**

For \(b \notin B\)

\[F^{-1}(t, b, F(s, b, x)) = x\]

**Lossiness:**

For \(b \in B\)

\[|\text{im}F(s, b, \cdot)| \leq 2^r\]

The first outputs of \(G_{ABN}(1^\lambda, B_0)\), and \(G_{ABN}(1^\lambda, B_1)\) are computationally indistinguishable.

Can be constructed from LTDFs
IND-SO-CCA Construction

KeyGen:

\[(s_0, t_0) \leftarrow \text{G}_{\text{LTDF}}(1^{\lambda}, \text{inj}) \]
\[(s_1, t_1) \leftarrow \text{G}_{\text{ABN}}(1^{\lambda}, \{1, \ldots, n\}) \]

\[\text{pk} = (s_0, s_1) \text{ and } \text{sk} = (t_0, t_1).\]

Encryption:

\[r_{\text{sig}} \leftarrow \text{coins(Sign)}, \ x \leftarrow X(vk, sk) = \text{G}(r_{\text{sig}}).\]

For a message \(m\), calculate:

\[
(F_{\text{LTDF}}(s_0, x), F_{\text{ABN}}(s_1, vk, x), h(x) \oplus m)\]

\[\text{sig} = \text{Sign}_{sk}(F_{\text{LTDF}}(s_0, x), F_{\text{ABN}}(s_1, vk, x), h(x) \oplus m),\]

output the ciphertext:

\[(vk, F_{\text{LTDF}}(s_0, x), F_{\text{ABN}}(s_1, vk, x), h(x) \oplus m, \text{sig})\]
IND-SO-CCA Construction

- **KeyGen:**

  \[
  (s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \quad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})
  \]

  \[pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).\]
IND-SO-CCA Construction

- **KeyGen:**

  \[(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, \text{inj}) \quad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})\]

  \[pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).\]

- **Encryption:**
IND-SO-CCA Construction

- **KeyGen:**
  \[(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \quad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})\]
  \[pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).\]

- **Encryption:**
  \[r^{sig} \leftarrow \text{coins}(\text{Sign}), \quad x \leftarrow X\]
  \[(vk, sk) = G(r^{sig}).\]
IND-SO-CCA Construction

- **KeyGen:**
  
  \[(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, \text{inj}) \quad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})\]

  \[pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).\]

- **Encryption:**
  
  \[r^{\text{sig}} \leftarrow \text{coins(Sign)}, \quad x \leftarrow X\]

  \[(vk, sk) = G(r^{\text{sig}}).\]

  For a message \(m\), calculate

  \[(F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m)\]

  \[\text{sig} = \text{Sign}_{sk}(F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m),\]

  output the ciphertext:

  \[(vk, F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m, \text{sig})\]
A SEM-SO-CCA Secure Construction
Intuition of our SEM-SO-CCA construction

To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.

There are difficulties:

An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.

Unduplicatable set selection [S99]

After we make $n$ simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.

The statistically hiding property of lossy encryption allows us to prove IND-SO security. Statistical NIZKs should allow us to prove IND-SO-CCA security.
Intuition of our SEM-SO-CCA construction

To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.

- There are difficulties:
  - An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
  - Unduplicatable set selection [S99]
  - After we make $n$ simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.
  - The statistically hiding property of lossy encryption allows us to prove IND-SO security. Statistical NIZKs should allow us to prove IND-SO-CCA security.
Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
  - Unduplicatable set selection [S99]
  - After we make $n$ simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.
  - The statistically hiding property of lossy encryption allows us to prove IND-SO security.
  - Statistical NIZKs should allow us to prove IND-SO-CCA security.
Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
  - Unduplicatable set selection [S99]
Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
  - Unduplicatable set selection [S99]
  - After we make $n$ simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.
Intuition of our SEM-SO-CCA construction

To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.

There are difficulties:

- An encryption query is actually a query for $n$ encryptions, so we need a NIZK which remains secure even after seeing $n$ simulated proofs.
- Unduplicatable set selection [S99]
- After we make $n$ simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.
- The statistically hiding property of lossy encryption allows us to prove IND-SO security.
- Statistical NIZKs should allow us to prove IND-SO-CCA security.
Statistical NIZKs [GOS06]

- Completeness: All true statements can be proven.
- Soundness: False statements (with witnesses to their falseness) cannot be proven.
- Zero-Knowledge: Nothing beyond the truth of the statement is revealed.
- Proof of Knowledge: There exists a simulator that can extract a witness from a valid proof.
- Honest-Prover State Reconstruction: There exists a simulator that can create a proof $P$ without a witness, then, given a witness $w$ can produce randomness $r$ such that $P$ appears to have been generated with $w$ and $r$. 

Brett Hemenway and Rafail Ostrovsky
Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
Statistical NIZKs [GOS06]

- **Completeness**: All true statements can be proven.
- **Soundness**: False statements (with witnesses to their falseness) cannot be proven.
Statistical NIZKs [GOS06]

- **Completeness**: All true statements can be proven.
- **Soundness**: False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge**: Nothing beyond the truth of the statement is revealed.
Statistical NIZKs [GOS06]

- **Completeness**: All true statements can be proven.
- **Soundness**: False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge**: Nothing beyond the truth of the statement is revealed.
- **Proof of Knowledge**: There exists a simulator that can extract a witness from a valid proof.
Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
- **Soundness:** False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge:** Nothing beyond the truth of the statement is revealed.
- **Proof of Knowledge:** There exists a simulator that can extract a witness from a valid proof.
- **Honest-Prover State Reconstruction:** There exists a simulator that can create a proof $P$ without a witness, then, given a witness $w$ can produce randomness $r$ such that $P$ appears to have been generated with $w$ and $r$. 
Tools

- Unduplicatable Set Selector
- SEM-SO-ENC secure encryption ($G$, $E$, $D$)
- Statistical NIZKs (Prover, Verifier, Ext, SR)
- Strongly Unforgeable One-Time Signatures (Sign, Ver)

Brett Hemenway and Rafail Ostrovsky
Tools

- Unduplicatable Set Selector $g$. 
Tools

- Unduplicatable Set Selector \( g \).
- SEM-SO-ENC secure encryption \( (G_{so}, E, D) \).
Tools

- Unduplicatable Set Selector $g$.
- SEM-SO-ENC secure encryption ($G_{so}, E, D$).
- Statistical NIZKs (Prover, Verifier, Ext, SR).
Tools

- Unduplicatable Set Selector $g$.
- SEM-SO-ENC secure encryption $(G_{so}, E, D)$.
- Statistical NIZKs (Prover, Verifier, Ext, SR).
- Strongly Unforgeable One-Time Signatures (Sign, Ver).
SEM-SO-CCA Construction

KeyGen: 
\((pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), (\sigma_i, \tau_i) \leftarrow Ext_{1^\lambda} \) for \( i \in L \)

\( pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \) and 
\( sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}) \).

Encryption: 
\( r_{sig} \leftarrow coins(Sign), r_0, r_1 \leftarrow coins(E), \{r_{nizk_i}\}_{i=1}^\ell \leftarrow coins(Prover). \)

\( (vk, sk) = G(r_{sig}) \)

For a message \( m \), calculate 
\( e_0 = E(pk_0, m, r_0), e_1 = E(pk_1, m, r_1) \)

set \( w = (m, r_0, r_1) \).

\( \pi = (\pi_1, ..., \pi_\ell) = (Prover(\sigma_i, (e_0, e_1), w), r_{nizk_i}) \)

\( \text{sig} = Sign(e_0, e_1, \pi) \), output the ciphertext: 
\( c = (vk, e_0, e_1, \pi, \text{sig}) \).
SEM-SO-CCA Construction

- **KeyGen:**

\[
(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L
\]

\[
pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).
\]
SEM-SO-CCA Construction

- **KeyGen:**

  \[(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L\]

  \[pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \text{ and } sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).\]

- **Encryption:**
SEM-SO-CCA Construction

- **KeyGen:**
  \[(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L\]
  \[pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \text{ and } sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).\]

- **Encryption:**
  \[r^{\text{sig}} \leftarrow \text{coins}(\text{Sign}), \ r_0, r_1 \leftarrow \text{coins}(E), \ \{r_i^{\text{nizk}}\}_{i=1}^\ell \leftarrow \text{coins}(\text{Prover}).\]
  \[(vk, sk) = G(r^{\text{sig}}).\]
SEM-SO-CCA Construction

KeyGen:

\((pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L\)

\(pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \text{ and } sk = (sk_0, sk_1, \{\tau_i\}_{i \in L})\).

Encryption:

\(r^{\text{sig}} \leftarrow \text{coins}(\text{Sign}), r_0, r_1 \leftarrow \text{coins}(E), \{r_i^{\text{nizk}}\}_{i=1}^\ell \leftarrow \text{coins}(	ext{Prover}).\)

\((vk, sk) = G(r^{\text{sig}}).\)

For a message \(m\), calculate

\(e_0 = E(pk_0, m, r_0), \quad e_1 = E(pk_1, m, r_1)\)

set \(w = (m, r_0, r_1)\).
SEM-SO-CCA Construction

**KeyGen:**

\[(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L\]

\[pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \text{ and } sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).\]

**Encryption:**

\[r^{\text{sig}} \leftarrow \text{coins}(\text{Sign}), r_0, r_1 \leftarrow \text{coins}(E), \{r_i^{\text{nizk}}\}_{i=1}^\ell \leftarrow \text{coins}(\text{Prover}).\]

\[(vk, sk) = G(r^{\text{sig}}).\]

For a message \(m\), calculate

\[e_0 = E(pk_0, m, r_0), \quad e_1 = E(pk_1, m, r_1)\]

set \(w = (m, r_0, r_1)\).

\[\overline{\pi} = (\pi_1, \ldots, \pi_\ell) = (\text{Prover}(\sigma_i, (e_0, e_1), w), r_i^{\text{nizk}})_{i \in g(vk)}\]

\[\text{sig} = \text{Sign}(e_0, e_1, \overline{\pi}),\]

output the ciphertext: \(c = (vk, e_0, e_1, \overline{\pi}, \text{sig}).\)
This construction is SEM-SO-CCA2 Secure
Our Results

ReRandomizable Encryption "is" Lossy Encryption

A framework for creating Lossy Encryption:

Applying the results of \[BHY09\] gives:

- Goldwasser-Micali
- El-Gamal
- Paillier / Damg˚ard-Jurik

The first proof that Paillier/Damg˚ard-Jurik is SEM-SO-ENC secure. This is the most efficient known SEM-SO-ENC cryptosystem.

Statistically Hiding-OT implies Lossy Encryption

PIR implies Lossy Encryption

Homomorphic Encryption implies Lossy Encryption

CCA2 Selective Opening Secure definitions and constructions

Constructions from statistically-hiding NIZKs in the simulation-based model

Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model

Brett Hemenway and Rafail Ostrovsky
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:

- Applying the results of [BHY09] gives:
  - Goldwasser-Micali
  - El-Gamal
  - Paillier/Damg˚ ard-Jurik
  - The first proof that Paillier/Damg˚ ard-Jurik is SEM-SO-ENC secure. This is the most efficient known SEM-SO-ENC cryptosystem.

- Statistically Hiding-OT implies Lossy Encryption
- PIR implies Lossy Encryption
- Homomorphic Encryption implies Lossy Encryption

- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model

Brett Hemenway and Rafail Ostrovsky
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik

- The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
- This is the most efficient known SEM-SO-ENC cryptosystem.

- Statistically Hiding-OT implies Lossy Encryption

- PIR implies Lossy Encryption

- Homomorphic Encryption implies Lossy Encryption

- CCA2 Selective Opening Secure definitions and constructions

- Constructions from statistically-hiding NIZKs in the simulation-based model

- Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
Our Results

▶ ReRandomizable Encryption “is” Lossy Encryption
  ▶ A framework for creating Lossy Encryption:
  ▶ Applying the results of [BHY09] gives:
    ▶ Goldwasser-Micali
    ▶ El-Gamal
    ▶ Paillier / Damgård-Jurik
  ▶ The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
▶ Statistically Hiding-OT implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
Our Results

▶ ReRandomizable Encryption “is” Lossy Encryption
  ▶ A framework for creating Lossy Encryption:
  ▶ Applying the results of [BHY09] gives:
    ▶ Goldwasser-Micali
    ▶ El-Gamal
    ▶ Paillier / Damgård-Jurik
  ▶ The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
▶ Statistically Hiding-OT implies Lossy Encryption
  ▶ PIR implies Lossy Encryption
  ▶ Homomorphic Encryption implies Lossy Encryption
▶ CCA2 Selective Opening Secure definitions and constructions
  ▶ Constructions from statistically-hiding NIZKs in the simulation-based model
Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model
Open Questions

▶ Can we construct an IND-CPA secure system that is not IND-SO secure?

▶ Can we remove the dependence on $n$ in the CCA constructions.

▶ What about receiver corruption?
Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?
Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?
- Can we remove the dependence on $n$ in the CCA constructions.
Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?
- Can we remove the dependence on $n$ in the CCA constructions.
- What about receiver corruption?
Open Question: Receiver Corruption

Recall: Sender Corruption Game
Open Question: Receiver Corruption

Sender Corruptions

\[ e_i = E(pk, m_i, r_i) \]
Open Question: Receiver Corruption

Sender Corruptions

\[ e_i = E(pk, m_i, r_i) \]
Open Question: Receiver Corruption

Receiver Corruptions

$e_i = E(pk_i, m_i, r_i)$
Open Question: Receiver Corruption

Receiver Corruptions

$e_i = E(pk_i, m_i, r_i)$
Open Question: Receiver Corruption

Receiver Corruptions

$e_i = E(pk_i, m_i, r_i)$
Thanks!